Learning to use Wavefront and SVE for Virtual Environments

Table of Contents:

Introduction

This document is intended to help you begin to use the tools available at Georgia Tech for creating interactive, real-time virtual environments (VEs). It will focus on three things: the use of the Wavefront Advanced Visualizer for model creation and editing, the use of the Simple Virtual Environment (SVE) Library for programming VE applications, and the use of the hardware available in the GVU lab for immersive virtual environments. There are other, more complete resources available, including the Wavefront manuals and the SVE manual. This guide will help you get started quickly and give you some background explaining how the various pieces fit together.

Related Documents:

- Wavefront tutorial
- SVE tutorial
- SVE User's Guide (Version 2.0)
- Instructions for using VE Hardware

Overview

Virtual environments (VEs) generally involve several diverse software and hardware components. A typical VE system might use the following:

- Graphics workstation
- Head-mounted display (HMD)
- Tracking devices for the user's head and hand
- VE system software which deals with tracker information, model maintenance, user interaction, and scene rendering
- Application software which uses the system software to realize particular application behaviors and properties
- Files describing the models, materials, and light sources in the VE

At Georgia Tech, many applications have been developed which use most or all of these components. The most important is the SVE library, a general-purpose VE system software library which allows rapid prototyping,

hardware independence, and run-time configuration. The GVU lab also has a large set of hardware resources which can be used to create immersive VE applications.

As an application developer, then, you must provide only the last two components in the list above. First, you will write an application program, which makes calls to the SVE library. For the simplest of virtual environments, this program could be as short as three lines of C code. SVE also allows complex behaviors and functionality, however, which can be included in the application program.

Second, the application developer must provide the 3D model to be used in the VE. If the model is very simple, it can be manually created using SVE's object file format. For most models, however, you will want to use a modeling program, such as Wavefront.

In the following sections, therefore, we will examine both Wavefront and SVE to help you quickly create a model and a simple application program. When these are ready, the final step is to learn how to use the hardware available in the lab. The final section of this document addresses that issue.

Wavefront

Before you can use Wavefront, you need to set some environment variables so it knows where to look for files. If you are using ksh (type `echo $SHELL` to find out which shell you're using), copy the following lines into your .profile-sgi file:

```
export WF_AV_DIR=/usr/aw/TAV4.3
export PATH=$PATH:$WF_AV_DIR/bin
export WF_MTL_LIB=$HOME/wave/colors.mtl
export WF_LGT_DIR=$HOME/wave
export WF_TEX_DIR=$HOME/wave
```

Otherwise, if you're using csh or tcsh, copy the following lines into your .login-sgi file:

```
setenv WF_AV_DIR /usr/aw/TAV4.3
setenv PATH ${PATH}:$WF_AV_DIR/bin
setenv WF_MTL_LIB $HOME/wave/colors.mtl
setenv WF_LGT_DIR $HOME/wave
setenv WF_TEX_DIR $HOME/wave
```

Now log out and log back in so the changes to your environment take effect.

Next, you should create a wave directory and copy the default materials file to it:

```
mkdir ~/wave
cp $WF_AV_DIR/dat/project/colors.mtl ~/wave
```

Finally, you can run the modeler or the property editor by typing the following commands:

```
model -ed
property
```

Jack Tumblin has created a Wavefront Tutorial that is quite useful. For the purposes of VE model creation, only the first section (on the use of the modeler) is necessary.

SVE

An outline from a short tutorial on SVE by Doug Bowman is available. Click here to see the tutorial.

See the SVE User's Manual for definitions, function descriptions, etc.

Important SVE topics:

- SVE basics: This section describes the event-based programming model, including application initialization, the interaction loop, and shutting down.
- Model maintenance: This section describes the object tree, where SVE maintains the current state of the environment model. Be sure to take a look at the section on the default object tree.
- Objects: The basic data structure in SVE is the SVE_object. This section describes how to create, load, save, and find objects in an SVE application.
- Object Appearance: Here we learn how to change the position, orientation, and geometry of objects in the SVE object tree.
- Events: Most SVE application code is simply a set of functions which respond to various events, like mouse clicks, key presses, or glove gestures. This section describes how callbacks work and how to use them in SVE.
- Other callbacks: This section describes callbacks that occur every frame (every time SVE redraws the scene), as well as default navigation techniques which are built-in callbacks to SVE.
- Sound: Sound feedback can add a great deal to a VE experience. SVE allows loading and playback of many sounds simultaneously. Read about it here.
- Servers: In order to run your application immersively, you need to use other processes which serve tracking, event, or audio information to/from your application. This section describes their use.

Hardware

The GVU lab in CRB has the following hardware available for use in immersive VE applications:

- Graphics Workstations:
  - SGI Max Impact (swift)
  - SGI Onyx2 Infinite Reality (elephant)
  - Windows NT PCs
- Head-mounted displays:
  - Virtual Research VR4
  - Virtual Research Eyegen3
  - Virtual Research Flight Helmet
  - Virtual I/O i-glasses
  - Virtuality HMD
- Tracker receivers:
  - Polhemus Isotrak II (two receivers)
  - Polhemus Fastrak (four receivers)
  - Polhemus InsideTrak (card in Windows NT machine)
- Tracker transmitters:
  - Polhemus standard transmitter

    - Polhemus Long Ranger transmitter
- Input devices:
    - Custom-built 3-button joysticks
    - Integrated stylus (pen) and tracker with button
    - CyberGlove
    - Custom-built pinch glove

For a brief, step by step set of instructions on setting up and using the common hardware, click here

---

Send questions, comments, or suggestions to:

bowman@cc.gatech.edu